

Attorney's Docket No.: 042390.P7512

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Patent Application of:)	
)	
Orna Etzion)	Examiner: Meonske, Tonia L.
)	
Application No.: 09/676,175)	Art Unit: 2183
)	
Filed: September 29, 2000)	
)	
For: A Method and Apparatus for)	
Generating an Expected Top of)	
Stack During Instruction)	
<u>Translation</u>)	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER 37 C.F.R. §1.131

Sir:

I, Orna Etzion, declare that:

1. I am the inventor of claims 1, 3-6, 8-11 and 13-15 of the above identified patent application.

2. Prior to June 16, 2000, I conceived the idea of method and apparatus for generating an expected top of stack during instruction translation as described and claimed in my application.

3. An Intel Invention disclosure, dated July 7 1999 (copy attached hereto as Exhibit A), which describes an embodiment of the invention, was prepared by myself as a submission to our legal team for consideration for filing a patent application. The invention disclosure describes the operation of generating an

expected top of stack during instruction translation, as is described and claimed in our application.

4. Sometime thereafter, the Intel patent legal team considered the invention disclosure and approved the invention disclosure for filing as an application in the United States.

5. Sometime thereafter, I traveled from Israel to the United States in the spring of 2000 to meet with our patent attorney to discuss the invention of the above identified patent application, as part of our continuous effort in preparing a draft of the above identified patent application.

6. A draft of the above-identified patent application was forwarded to myself, via the email from John Ward on June 24, 2000. I received and reviewed the draft of the patent application, and provided my feedback on the draft on July 3, 2000. (Copies of the emails are attached hereto as Exhibit B)

7. Following subsequent back and forth communications between myself located in Israel and the attorney located in the California, I believe the above-identified patent application was filed thereafter with the PTO on September 29, 2000.

8. We declare, to the best of our knowledge, all statements made in this document are true, and that all statements made on information are believed to be true; and further, that these statements were made with the knowledge that willful false statements are punishable by fine or imprisonment, or both, under § 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the above-identified patent application or any patent issued thereon.

Date: February 1, 2005 _____

ML
Comm / MPG / MPL file

INTEL INVENTION DISCLOSURE

INTEL CONFIDENTIAL

JUL - 7 1999

LEGAL ID# 12073DATE: July 7, 1999

It is important to provide accurate and detailed information on this form. The information will be used to evaluate your invention for possible filing as a patent application. When completed, please return this form to the Legal Department at RN4-01. If you have any questions regarding this form or to whom it should be forwarded, please call 765-1369, 696-2851 or 554-3996.

1. Inventor(s):

Name: Orna Etzion SS# N/A
Empl. No. 10122359 Dept.# 6985 Phone 4865-5720 M/S: IDC-1 D
Home Address: 5 Kariv st. Haifa, Israel
Citizenship: Israel Supervisor* Yaron Sheffer M/S: IDC-1 D
Group Name: MPL Division Name: MPG

RECEIVED

JUL 08 1999

Name: ? SS# N/A PATENT DATABASE GROUP
Empl. No. Dept.# 6985 Phone ? M/S: IDC-1 D **INTEL LEGAL TEAM**
Home Address:
Citizenship: Israel Supervisor* Yaron Sheffer Phone 4865-5759 M/S: IDC-1 D
Group Name: ML Division Name: MPG

2. Title of Invention: A method for efficiently maintaining synchronization of a simulated circular-stack of registers during binary translation.

3. Stage of development, i.e. % complete, and relation of technology to the following product/process:
The technique has been implemented in a dynamic IA32→IA64 binary translator, which is currently a research project, for floating-point stack simulation.

4. (a) Has a description of your invention been, or will it shortly be, published outside Intel:

NO: YES: X DATE WAS OR WILL BE PUBLISHED: 10/99

If YES, was the manuscript submitted for pre-publication approval? YES: X NO:

(b) Has your invention been used/sold or planned to be used/sold by Intel or others?

NO: YES: X DATE WAS OR WILL BE SOLD: may be used in future implementations of IA64, not yet on plan of record.

5. If invention conceived, or constructed during performance of a government or third party contract, please check here and give the contract name and number .

6. Please attach a page to this form, DATED AND SIGNED BY ONE INVENTOR (PREPARER), to provide an abstract of your invention, and include the following information in your abstract:

- (a) State general purpose(s) of your invention;
- (b) Describe advantage(s) of your invention over what is done now;
- (c) Describe essential element(s) or key to your invention; and
- (d) Value of your invention to Intel (how will it be used?).

*HAVE YOUR SUPERVISOR READ, DATE AND SIGN COMPLETED FORM

DATE: ? SUPERVISOR: Yaron Sheffer

BY THIS SIGNING, I (SUPERVISOR) ACKNOWLEDGE THAT I HAVE READ AND UNDERSTAND THIS DISCLOSURE, AND RECOMMEND THAT THE HONORARIUM BE PAID.

General purpose of the invention

The purpose of this invention is to efficiently maintain synchronization of a simulated circular register stack. The invention may be valuable for binary translation, from source computer architecture that contains such a stack, to a target architecture that supports a flat register file. The invention may be used in dynamic or static binary-translators, as well as in architectural simulators or virtual-machine implementations using similar, code-generation-based, techniques. In particular, the invention provides a significant performance advantage when translating Intel Architecture floating-point code to any other architecture.

Advantages of the invention over what is done now

The invention is significantly faster than any known alternative.

Emulating a stack rotation by multiple move operations need to perform those moves for any stack push or pop. The number of the required moves per occurrence is the size of the stack, and they contain a lot of internal dependencies. In the proposed invention, the rotation moves are performed only on extremely rare cases.

Emulating a stack in memory suffers from a great load-store overhead, which the proposed invention avoids.

Essential elements or key to the invention

The following section demonstrates the key elements of the invention using, as an example, an IA32→IA64 binary translator. The relevant aspect is the emulation of IA32 floating-point (FP) register stack, using the flat FP register-file of an IA64 target machine.

References to the eight physical FP-registers of the Intel IA32 architecture are always stack-relative. The mapping between stack-relative references and physical registers changes dynamically. For example, the physical registers corresponding to ST(0) before and after executing an FLD instruction are different, since FLD pushes a value onto the FP-stack.

However, in the vast majority of practical cases, multiple run-time entries to the same code block repeat the same stack-depth at entrance. Speculating the state at the entry point allows an effective static mapping between any IA32 FP register-references in the block and the corresponding IA64 FP-registers. To take advantage of such a speculative approach, the following mechanisms are supported:

1. Stack depth speculation – effectively guessing the run-time stack state at all or almost all entries to the block. The speculation is done prior to the block translation. Dynamic translator uses the 1st run-time entry state (which is already known when the block is reached). Static translator has to perform code analysis and walk-through to predict the entrance state effectively.
2. Tracking the speculation realization – keeping the actual run time stack state and verifying that the speculative assumption (taken at the translation of the block) is indeed true at each run-time entry. The actual stack depth is updated at the end of the block execution, which is a single operation that reflects the overall effect of the entire block. If the block is balanced (same number of pushes and pops), this code is eliminated. At the beginning of each block, a checking code is executed, that compares the assumed (speculated) stack depth with the actual one.
3. Recovery mechanism – ensure correct operation when the check fails. The recovery is achieved by actual rotation (copy of register values), so the actual top-of-stack moves to fit the expected one. The block code remains as is. This method of recovery ensures that the penalty does not propagate: When control is transferred to the next block, the correction is already done, and the stack-depth expected by the next block matches the actual depth.

Note: This invention disclosure does not describe how stack exception conditions are detected. The solution to that problem is covered by another patent disclosure.

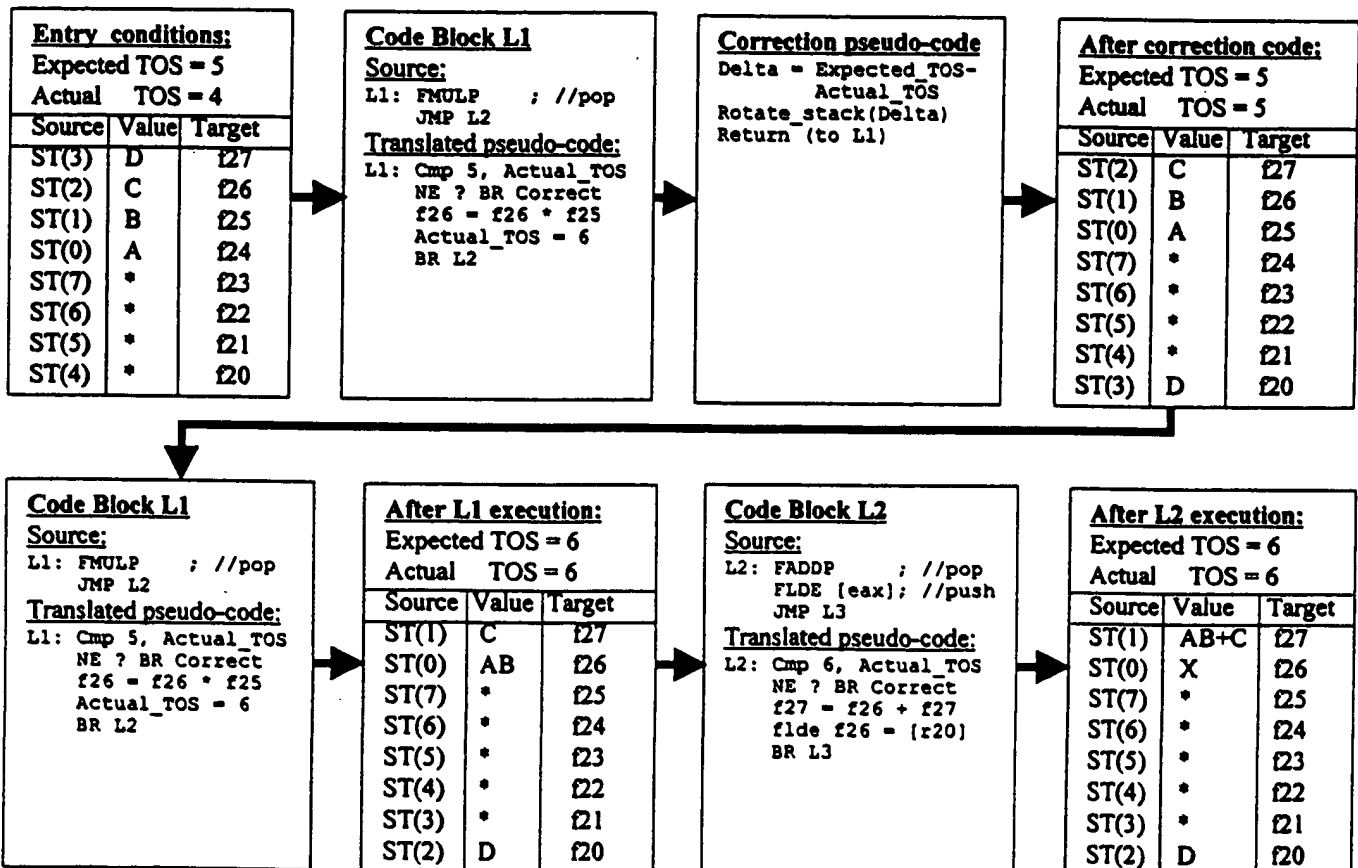
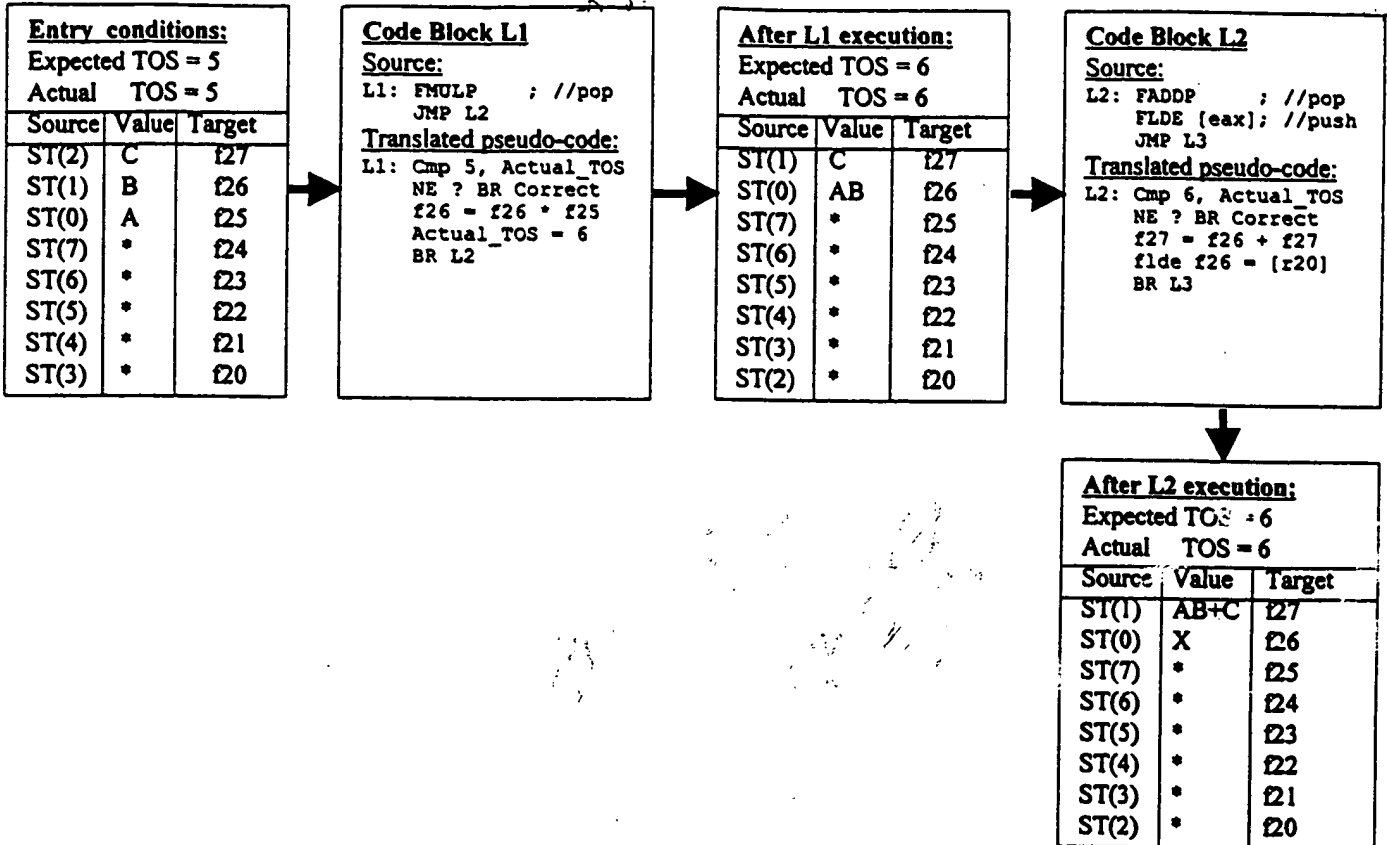
Example

The example in the following page consists of 2 very simple floating-point blocks. It shows the behavior of the translation mechanism at the regular case (when the expected Top-Of-Stack equals the actual one), and on the special case (when they are different). Note that L2 block is balanced, hence no update of the actual TOS value is done at its epilogue. Also note that the correction done for L1 (on the special case) does not affect the normal flow at L2. The Actual TOS value is best held in a global integer register (but not necessarily).

As already stated, although the example refers to IA32→IA64 translation, the invention principles are applicable to any other case of emulating a rotating stack by a static register file.

Value of the invention to Intel: how will it be used?

This invention is valuable to Intel because it can be used to significantly speed up the floating-point performance of IA32→IA64 dynamic binary translation. Such a project currently exists as a research project, but the technology is expected to eventually enter a commercial product of strategic importance to Intel.

Example



"Etzion, Orna" <orna.etzion@intel.com> on 07/03/2000 03:44:32 AM

To: John Ward/Bstz
cc: "Etzion, Orna" <orna.etzion@intel.com>
Subject: RE: Patent applications

Hi John,

Here are my comments on the draft:

1. page 3 line 8: the overhead is reduced (not eliminated), especially for the "on the fly case" where the translation itself is part of the overhead.
 2. page 4: waiting for the drawings to be faxed.
 3. page 5 line 19: in "programs" I understand that you mean the programs who are being emulated/translated.
 4. page 5 line 23: Stacks may keep ... is only an example so maybe should be mentioned under the for example (in line 24).
- page 6 line 1: I did not like the "used in this way". I did not understand what you mean by this.

I understood that from page 6 line 6 to page 7 line 6 you describe what it means to use the stack in the original program in an architecture that has a HW built in stack. Following comments are based on this undersatnding.

page 6 line 6: What is missing in this paragraph is the explanation that the instructions that refer to the stack are refering to relative to TOS based operands. They do not refer to ST0, ST1 etc. but will always refer to TOS, TOS-1 etc. In the example in terms of the instructions there will be no difference between the 1st instruction which will push the element into ST0 and the 2nd instruction which pushes the element into ST1. Both will be pushing into the TOS. It is the HW which maintains the identity of the current TOS (knowing which of the physical entries (ST0-ST4) it currently is.

page 6 line 14: the TOS is not passed from one BB to the next in the original programs. The original program expects the HW to maintain TOS. The important point in the paragraphs that discuss BBs is that it is possible (in the original program) to enter a BB when TOS is a different physical register. The original code will work ok, because the HW will maintain the correct TOS.

page 7 line 5: Again, the original program does not care that the TOS can change from one execution of the BB to the next. The HW will ensure that the instructions will use and set the appropriate physical registers.

I undersstood that from page 7 line 9 to line 24 you describe the general

Regards, Orna

-----Original Message-----

From: John Ward [mailto:John_Ward@bstz.com]

Sent: Saturday, June 24, 2000 3:00 AM

To: orna.etzion@intel.com

Subject: Patent applications

Orna, enclosed is a rough first draft of the patent application originally entitled "maintaining synchronization of a simulated circular-stack of registers during binary translation". Please send me you fax number so that I can fax the figures to you. I need to file the application June 30th. Please let me know when is convenient to discuss your comments/revisions on the draft.

Regards,
-John

(See attached file: P7512 Patent application.ver1.doc)